# Trilha de Teoria da Computação

Para receber um certificado dessa trilha, o aluno deve cursar

- as obrigatórias de pelo menos 2 módulos e
- pelo menos 7 disciplinas da trilha.

# Trilha de Teoria da Computação

Para receber um certificado dessa trilha, o aluno deve cursar

- as obrigatórias de pelo menos 2 módulos e
- pelo menos 7 disciplinas da trilha.

Três módulos:

- Algoritmos
- Otimização (Marcel; 11 de abril)
- Matemática discreta (Yoshi; 13 de junho)

# Trilha de Teoria da Computação

Para receber um certificado dessa trilha, o aluno deve cursar

- as obrigatórias de pelo menos 2 módulos e
- pelo menos 7 disciplinas da trilha.

Três módulos:

- Algoritmos
- Otimização (Marcel; 11 de abril)
- Matemática discreta (Yoshi; 13 de junho)

Hoje: Módulo Algoritmos

# Escalonamento de máquinas idênticas

Dados: $m$ máquinas
$t$ tarefas
duração $d[i]$ da tarefa $i$ $\quad (i = 1, \ldots, t)$

# Escalonamento de máquinas idênticas

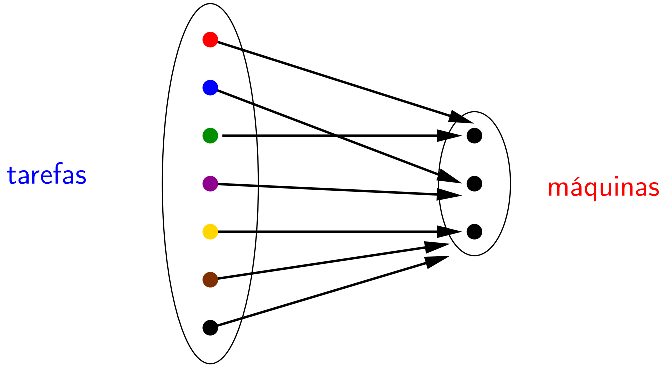
Dados: $m$ máquinas
$t$ tarefas
duração $d[i]$ da tarefa $i$ $\quad (i = 1, \ldots, t)$

um escalonamento é
uma partição $\{M[1], \ldots, M[m]\}$ de $\{1, \ldots, t\}$

# Escalonamento de máquinas idênticas

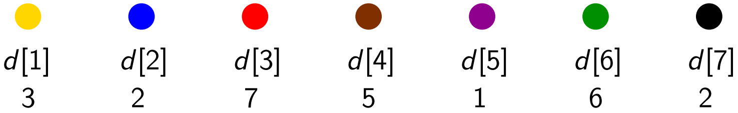
Dados: $m$ máquinas
$t$ tarefas
duração $d[i]$ da tarefa $i$ $\quad (i = 1, \ldots, t)$

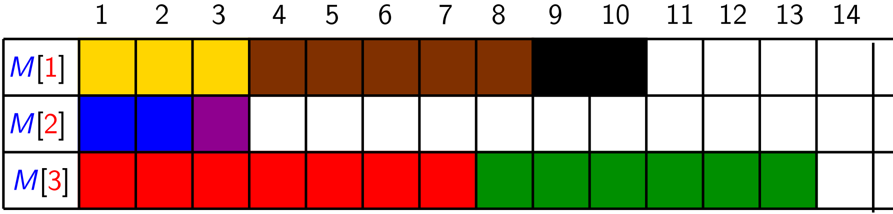
um escalonamento é
uma partição $\{M[1], \ldots, M[m]\}$ de $\{1, \ldots, t\}$



# Exemplo 1

$m = 3 \qquad t = 7$



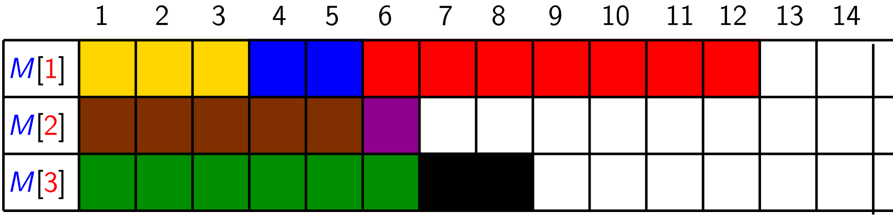




$\{\{1, 4, 7\}, \{2, 5\}, \{3, 6\}\} \Rightarrow$ Tempo de conclusão $= 13$

# Exemplo 2

$m = 3 \quad t = 7$

| $d[1]$ | $d[2]$ | $d[3]$ | $d[4]$ | $d[5]$ | $d[6]$ | $d[7]$ |
|---|---|---|---|---|---|---|
| 3 | 2 | 7 | 5 | 1 | 6 | 2 |

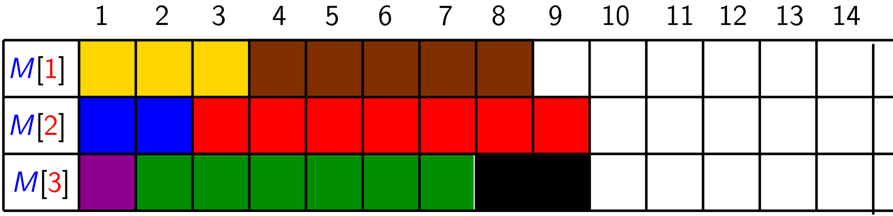



$\{\{1, 2, 3\}, \{4, 5\}, \{6, 7\}\} \Rightarrow$ Tempo de conclusão $= 12$

# Problema

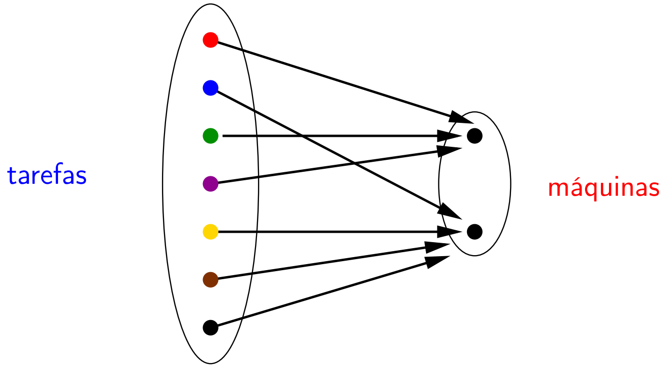
Encontrar um escalonamento com tempo de conclusão mínimo.

| $d[1]$ | $d[2]$ | $d[3]$ | $d[4]$ | $d[5]$ | $d[6]$ | $d[7]$ |
|---|---|---|---|---|---|---|
| 3 | 2 | 7 | 5 | 1 | 6 | 2 |




$\{\{1,4\},\{2,3\},\{5,6,7\}\} \Rightarrow$ Tempo de conclusão $= 9$

# NP-difícil mesmo para $m = 2$




Algoritmo: testa todo $M[1] \subseteq \{1, \ldots, t\}$ e escolhe melhor

$2^t$ subconjuntos $\Rightarrow$ exponencial

NP-difícil $\Rightarrow$ é improvável que exista algoritmo polinomial
que resolva o problema (se existir, $P = NP$)

# Algoritmo de Graham

Atribui, uma a uma, cada tarefa à máquina menos ocupada.

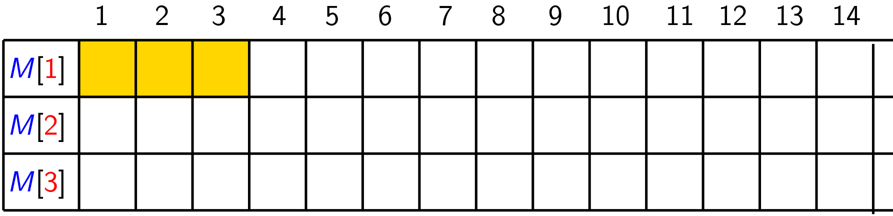
| $d[1]$ | $d[2]$ | $d[3]$ | $d[4]$ | $d[5]$ | $d[6]$ | $d[7]$ |
|---|---|---|---|---|---|---|
| 3 | 2 | 7 | 5 | 1 | 6 | 2 |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $M[1]$ | | | | | | | | | | | | | | |
| $M[2]$ | | | | | | | | | | | | | | |
| $M[3]$ | | | | | | | | | | | | | | |

# Algoritmo de Graham

Atribui, uma a uma, cada tarefa à máquina menos ocupada.

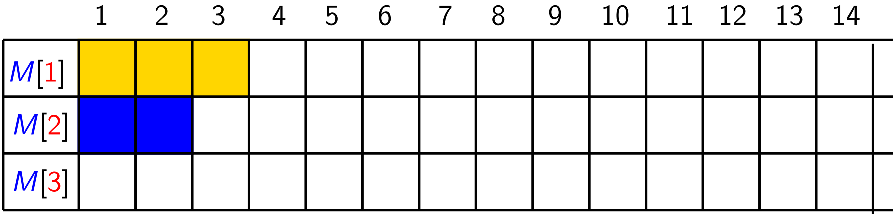
| $d[1]$ | $d[2]$ | $d[3]$ | $d[4]$ | $d[5]$ | $d[6]$ | $d[7]$ |
|---|---|---|---|---|---|---|
| 3 | 2 | 7 | 5 | 1 | 6 | 2 |



# Algoritmo de Graham

Atribui, uma a uma, cada tarefa à máquina menos ocupada.

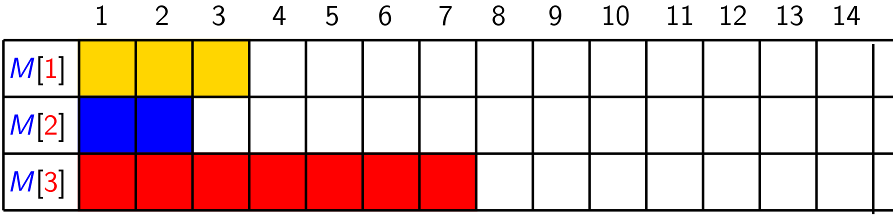
| $d[1]$ | $d[2]$ | $d[3]$ | $d[4]$ | $d[5]$ | $d[6]$ | $d[7]$ |
|---|---|---|---|---|---|---|
| 3 | 2 | 7 | 5 | 1 | 6 | 2 |



# Algoritmo de Graham

Atribui, uma a uma, cada tarefa à máquina menos ocupada.

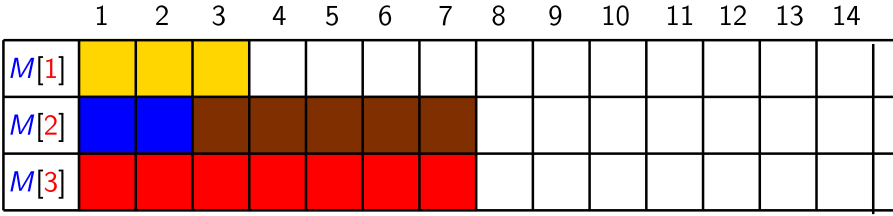
| $d[1]$ | $d[2]$ | $d[3]$ | $d[4]$ | $d[5]$ | $d[6]$ | $d[7]$ |
|---|---|---|---|---|---|---|
| 3 | 2 | 7 | 5 | 1 | 6 | 2 |



# Algoritmo de Graham

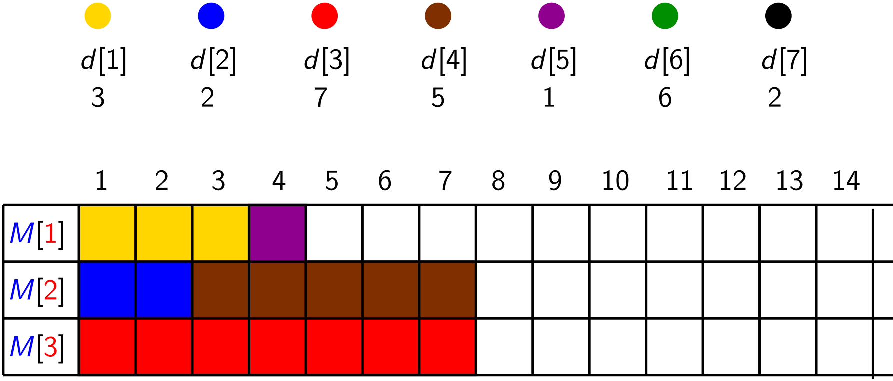
Atribui, uma a uma, cada tarefa à máquina menos ocupada.

| $d[1]$ | $d[2]$ | $d[3]$ | $d[4]$ | $d[5]$ | $d[6]$ | $d[7]$ |
|---|---|---|---|---|---|---|
| 3 | 2 | 7 | 5 | 1 | 6 | 2 |



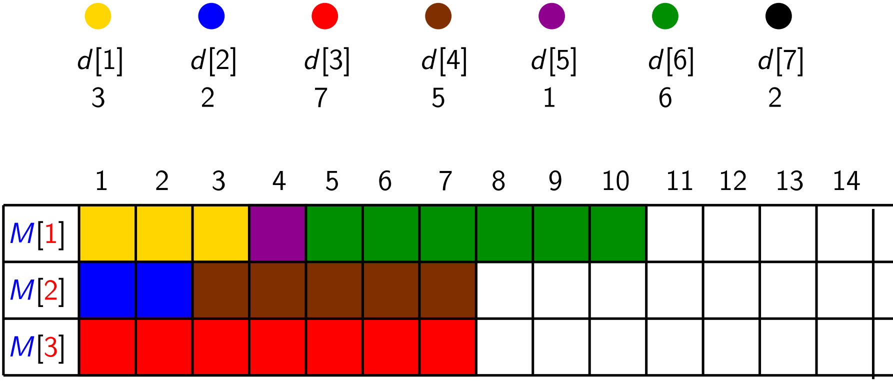
# Algoritmo de Graham

Atribui, uma a uma, cada tarefa à máquina menos ocupada.



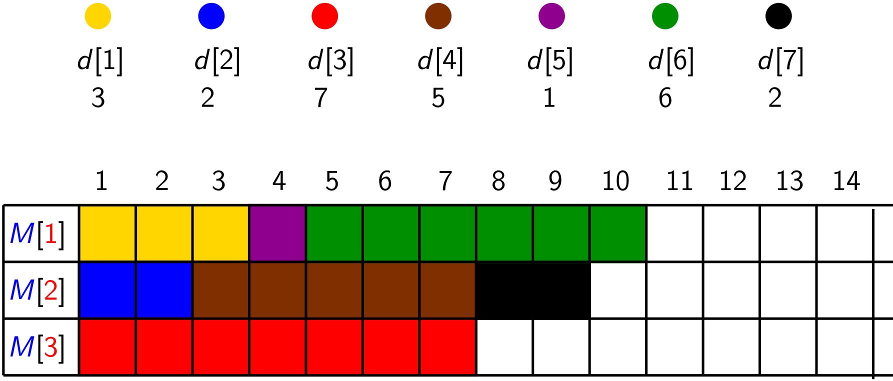
# Algoritmo de Graham

Atribui, uma a uma, cada tarefa à máquina menos ocupada.



# Algoritmo de Graham

Atribui, uma a uma, cada tarefa à máquina menos ocupada.



# Escalonamento-Graham

Recebe números inteiros positivos $m$ e $t$ e um vetor $d[1\,..\,t]$ e devolve um escalomento de $\{1, \ldots, t\}$ em $m$ máquinas.

ESCALONAMENTO-GRAHAM $(m, t, d)$

```
para j ← 1 até m faça
    M[j] ← ∅
    T[j] ← 0
para i ← 1 até t faça
    seja k tal que T[k] é mínimo
    M[k] ← M[k] ∪ {i}
    T[k] ← T[k] + d[i]
devolva {M[1], ..., M[m]}
```

# Delimitações para OPT

$\mathrm{OPT}$ = menor tempo de conclusão de um escalonamento

- Duração da tarefa mais longa:

$$\mathrm{OPT} \geq \max\{d[1], d[2], \ldots, d[t]\}$$

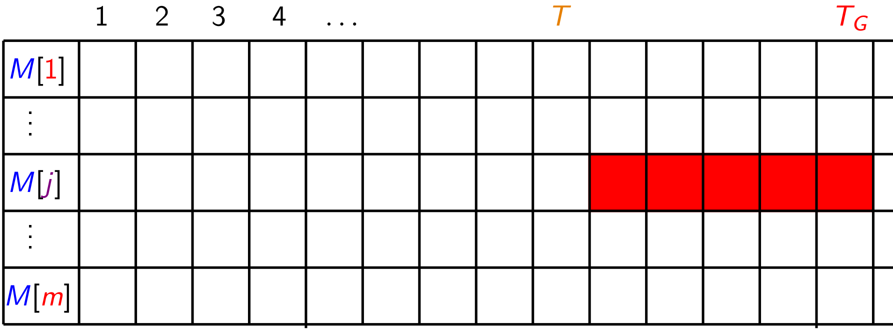
- Distribuição balanceada:

$$\mathrm{OPT} \geq \frac{d[1] + d[2] + \cdots + d[t]}{m}$$

# Qualidade do escalonamento

tarefa $i$ é executada na máquina $j$ a partir do instante $T$

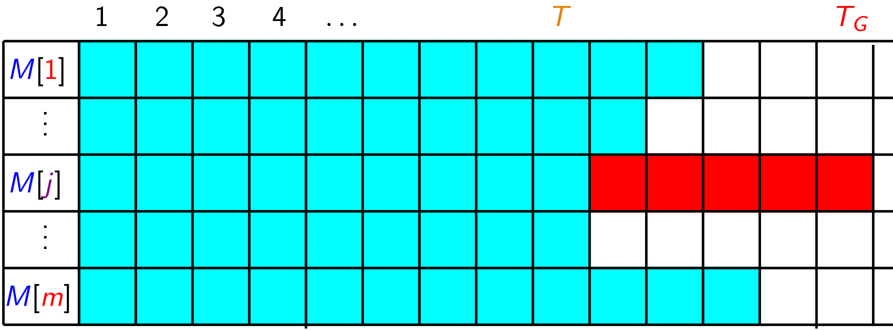
$T_G$ = conclusão do escalonamento obtido pelo algoritmo



# Qualidade do escalonamento

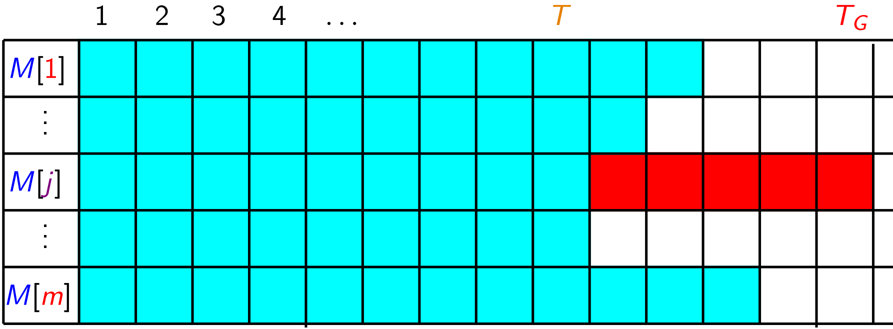
tarefa $i$ é executada na máquina $j$ a partir do instante $T$

$T_G$ = conclusão do escalonamento obtido pelo algoritmo




$$T \cdot m < d[1] + \cdots + d[t] \quad \Rightarrow \quad T < \frac{d[1] + \cdots + d[t]}{m}$$

# Qualidade do escalonamento




$$\begin{aligned}
T_G &= T + d[i] \\
&< \frac{d[1] + \cdots + d[t]}{m} + d[t] \\
&\le \frac{d[1] + \cdots + d[t]}{m} + \max\{d[1], \ldots, d[t]\} \\
&\le \mathrm{OPT} + \mathrm{OPT} = 2\,\mathrm{OPT}
\end{aligned}$$

# Conclusão

$\text{ESCALONAMENTO-GRAHAM}\ (m, t, d)$

```
para j ← 1 até m faça
    M[j] ← ∅
    T[j] ← 0
para i ← 1 até t faça
    seja k tal que T[k] é mínimo
    M[k] ← M[k] ∪ {i}
    T[k] ← T[k] + d[i]
devolva {M[1], . . . , M[m]}
```

O algoritmo ESCALONAMENTO-GRAHAM é uma $2$-aproximação.

# Módulo Algoritmos

- MAC0328 Algoritmos em Grafos
- MAC0414 Autômatos, Computabilidade e Complexidade

# Módulo Algoritmos

- MAC0328 Algoritmos em Grafos
- MAC0414 Autômatos, Computabilidade e Complexidade

- MAC0325 Otimização Combinatória
- MAC0327 Desafios de Programação I
- MAC0345 Desafios de Programação II
- MAC0331 Geometria Computacional
- MAC0450 Algoritmos de Aproximação
- MAC0336 Criptografia para Segurança de Dados
- MAC0465 Biologia Computacional
- MAC0466 Teoria dos Jogos Algorítmica
- MAC0333 Armazenamento e Recuperação de Informação
- MAC0xxx Estruturas de Dados Avançadas

# Módulo Matemática Discreta

- MAT0206 Análise Real
- MAT0264 Aneis e corpos
- MAC0320 Introdução à Teoria dos Grafos

# Módulo Matemática Discreta

- MAT0206 Análise Real
- MAT0264 Aneis e corpos
- MAC0320 Introdução à Teoria dos Grafos

- MAT0311 Cálculo Diferencial e Integral V
- MAT0225 Funções Analíticas
- MAT0313 Álgebra III
- MAT0234 Medida e Integração
- MAE0221/224 Probabilidade I e II
- MAE0228 Noções de Probabilidade e Processos Estocásticos
- MAE0326 Aplicações de Processos Estocásticos
- MAC0414 Autômatos, Computabilidade e Complexidade
- MAC0775/776 Métodos Probabilísticos em Combinatória e em Teoria da Computação I e II
- MAC0690/692 Tópicos em Combinatória Contemporânea I e II
- MAC0436/556 Tópicos de Matemática Discreta I e II
- MAC0691 Tópicos na Teoria Algébrica dos Grafos.

# Módulo Otimização

- MAC0315 Otimização Linear
- MAC0325 Otimização Combinatória

# Módulo Otimização

- MAC0315 Otimização Linear
- MAC0325 Otimização Combinatória

- MAC0300 Métodos Numéricos da Álgebra Linear
- MAC0473 Otimização Inteira
- MAC0450 Algoritmos de Aproximação
- MAC0427 Programação Não Linear
- MAC0419 Métodos de Otimização em Finanças
- MAC0452 Tópicos de Otimização Combinatória I
- MAC0552 Tópicos de Otimização Combinatória II
- MAC0461 Introdução ao Escalonamento e Aplicações
- MAC0418 Tópicos Especiais de Programação Matemática

# MAC0328 Algoritmos em Grafos

Objetivos: Estudar algoritmos para problemas fundamentais em grafos.

# MAC0328 Algoritmos em Grafos

Objetivos: Estudar algoritmos para problemas fundamentais em grafos.

Programa:

- Conexão de grafos: componentes, grafos biconexos.
- Digrafos fortemente conexos
  (alg. Kosaraju-Sharir, alg. Tarjan).
- Emparelhamentos máximos em grafos bipartidos.
- Emparelhamentos em grafos arbitrários (alg. Edmonds).
- Fluxo máximo (alg. Ford-Fulkerson).
- Coloração de vértices.
- Circuitos hamiltonianos.
- Tópicos opcionais:
  link analysis, network analysis, redes aleatórias.

# MAC0328 Algoritmos em Grafos

Objetivos: Estudar algoritmos para problemas fundamentais em grafos.

Programa:

- Conexão de grafos: componentes, grafos biconexos.
- Digrafos fortemente conexos
  (alg. Kosaraju-Sharir, alg. Tarjan).
- Emparelhamentos máximos em grafos bipartidos.
- Emparelhamentos em grafos arbitrários (alg. Edmonds).
- Fluxo máximo (alg. Ford-Fulkerson).
- Coloração de vértices.
- Circuitos hamiltonianos.
- Tópicos opcionais:
  link analysis, network analysis, redes aleatórias.

Bibliografia principal:

Sedgewick & Wayne, Algorithms, 4a ed., 2011.

# MAC0414 Autômatos, Computabilidade e Complexidade

Objetivos: Estudo de vários formalismos para computação e algoritmos e as limitações de certas formas de computação.

# MAC0414 Autômatos, Computabilidade e Complexidade

Objetivos: Estudo de vários formalismos para computação e algoritmos e as limitações de certas formas de computação.

Programa:

- Palavras, linguagens, operações sobre linguagens.
- Linguagens regulares.
- Autômatos finitos determinísticos e não determinísticos.
- Teorema de Kleene.
- Autômatos reduzidos.
- Modelos de computação; máquinas de Turing.
- Tese de Church.
- Redutibilidade e problemas indecidíveis.
- Complexidade, problemas decidíveis em tempo polinomial.
- Não-determinismo versus determinismo.
- Redutibilidade e problemas NP-completos.
- Teorema de Cook-Levin.

# MAC0414 Autômatos, Computabilidade e Complexidade

Objetivos: Estudo de vários formalismos para computação e algoritmos e as limitações de certas formas de computação.

Programa:

- Palavras, linguagens, operações sobre linguagens.
- Linguagens regulares.
- Autômatos finitos determinísticos e não determinísticos.
- Teorema de Kleene.
- Autômatos reduzidos.
- Modelos de computação; máquinas de Turing.
- Tese de Church.
- Redutibilidade e problemas indecidíveis.
- Complexidade, problemas decidíveis em tempo polinomial.
- Não-determinismo versus determinismo.
- Redutibilidade e problemas NP-completos.
- Teorema de Cook-Levin.

Bibliografia principal:

Sipser, Introduction to the Theory of Computation, 3a ed., 2012.

# Módulo Algoritmos

Demais disciplinas:

- MAC0325 Otimização Combinatória
- MAC0327 Desafios de Programação I
- MAC0345 Desafios de Programação II
- MAC0331 Geometria Computacional
- MAC0450 Algoritmos de Aproximação
- MAC0336 Criptografia para Segurança de Dados
- MAC0465 Biologia Computacional
- MAC0466 Teoria dos Jogos Algorítmica
- MAC0333 Armazenamento e Recuperação de Informação
- MAC0xxx Estruturas de Dados Avançadas

# Trilha de teoria

Esse semestre:

- MAC0315 Otimização Linear
- MAC0320 Introdução à Teoria dos Grafos
- MAC0327 Desafios de Programação I
- MAC0336 Criptografia para Segurança de Dados
- MAC0xxx Estruturas de Dados Avançadas

Próximo semestre:

- MAC0328 Algoritmos em Grafos
- MAC0333 Armazenamento e Recuperação de Informação
- MAC0345 Desafios de Programação II
- MAC0427 Programação Não Linear
- MAC0680 Tópicos em Combinatória Contemporânea I

  E uma dentre
- MAC0325 Otimização Combinatória
- MAC0331 Geometria Computacional
- MAC0450 Algoritmos de Aproximação